Direct Equivalence Testing

Rohit Pagariya, John Regehr
School of Computing, University of Utah
{pagariya, regehr}@cs.utah.edu

Introduction

- Testing embedded software is difficult.
- Further complicated by presence of memory and type safety errors in software.
- Compiler contain various known bugs. Developers are skeptical to upgrade the compilers.
- **Is your embedded software affected by memory safety and compilation errors?**

Our solution

Direct Equivalence Testing:

Equivalence testing is checking for violation of equivalence in the given equivalent programs.

Research Contributions

- Solving the problems of **interrupt driven concurrency, compiler optimizations and memory mapped IO** by applying the technique of equivalence testing to real world embedded software.

Why care?

- Embedded systems and software are becoming an integral part of our lives in the 21st century.

- Needless to say, bugs in embedded software could potentially be fatal.

Methodology

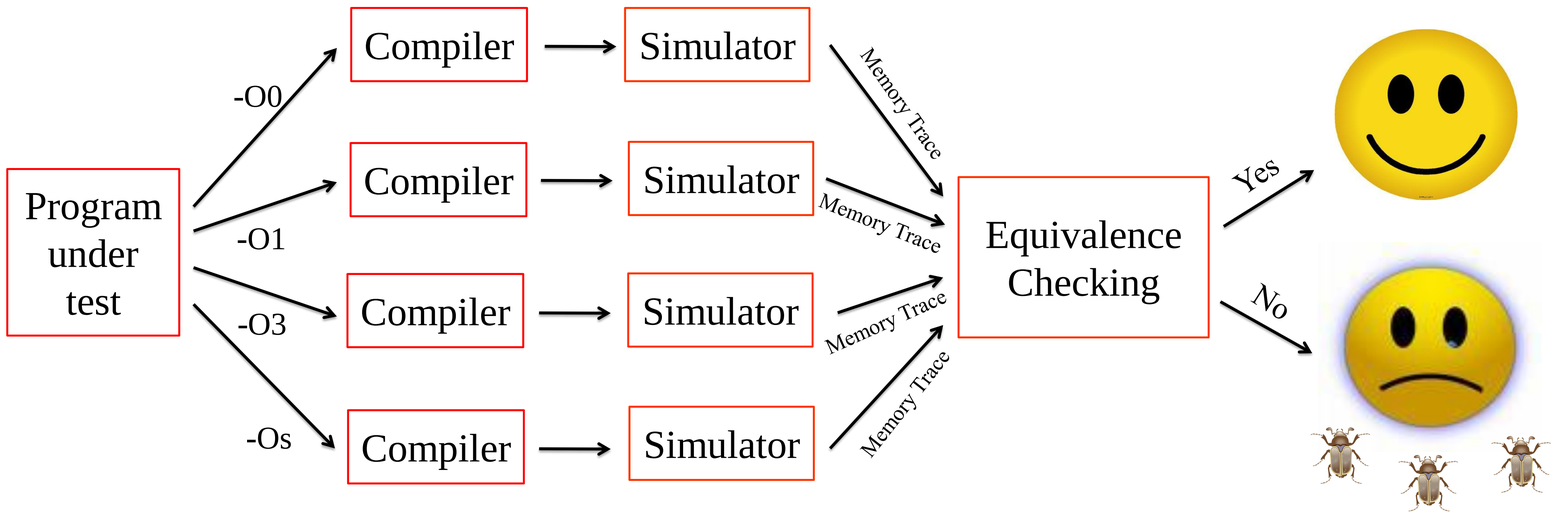

Figure 1. Direct Equivalence Testing

The memory traces generated by a computation indicate the input and output machine state of the computation. The first read reference to a memory location is part of the input state. The last write reference to a memory location is part of the output state.

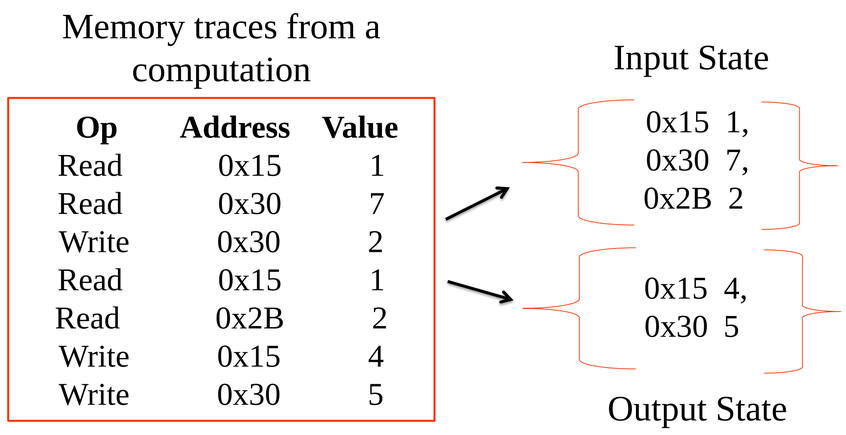

Figure 2. Extrapolating the machine states from memory traces

Equivalent input machine states of a computation imply output machine states must be equivalent. Checking is performed at known equivalent points in a program.

Sample Bugs

- **Correctness error in msp430-gcc**:

```
int64_t foo = 0;
int64_t bar = 123456789123456;
foo = (bar >> 40) ;
```

The expected value was 112 but garbage value was returned.

- **Correctness error in llvm-msp430**:

```
int32_t foo = 0, bar = 1;
return ((!foo & 0x00) != bar);
```

The expected value was 1 but 0 was returned.

- **Programming error (out of bounds access) in MultihopOscilloscope application of TinyOS:**

```
int readings[5];
readings[5] = 1;
```

- **Portability error in msp430-gcc**:

ADC related memory in msp430 microcontrollers is only word addressable. Byte accesses to this memory give unpredictable results. **The C standard doesn't specify if it is the compiler writer or compiler user who is responsible for ensuring compliance** in such situations.

Results

- **Direct Equivalence Testing** can detect any error – compiler or application – that results in different values being stored to memory (RAM).
- Types of errors detected are:
  - **Compiler Errors:**
    - Correctness errors
    - Volatile qualifier related errors
  - **Programming errors:**
    - Out of bounds accesses
    - Stack overflow
    - Use of uninitialized variables
  - **Portability errors**